Amendments to the Specification

Please replace the below listed paragraphs in the specification with the following paragraphs:

Fourth full paragraph beginning on line 23 of page 1 with:

With the advent of such useful software applications, it has become popular to develop and use a variety of intelligent documents where such documents may point to and download via a software application helpful solutions that enhance the performance of such documents. For example, a ~~resume~~ resumé document may be developed as a template with open fields for each section of the document such as the "experience" section or the "education" section. The document may point to a solution that provides helpful functionality. For example, after downloading a particular solution, a user of the document may be provided a tutorial on completing ~~resumes~~ resumés, or the user may be provided helpful information as the user focuses an insertion point on certain fields of the document. For example, if the user focuses on the education section of the document, the user may be provided helpful information as to how many education entries are suggested.

First full paragraph beginning on line 1 of page 5 with:

Referring now to the drawings in which like numerals refer to like parts or components through the several Figures. Fig. 1 is a simplified block diagram illustrating a client-server architecture for use in conjunction with an embodiment of the present invention. All components and files necessary to operate or fully implement the functionality or solutions available to a software application ~~110~~ 100 are identified and are assembled on a local library of software components in the schema library 105. For a detailed description of the use of an operation of Namespace or schema libraries, see U.S. Patent Application entitled "System and Method for Providing Namespace Related Information," U.S. Serial No. 10/184,190, filed June 27, 2002, and U.S. Patent Application entitled "System and Method for Obtaining and Using Namespace Related Information for Opening XML Documents," U.S. Serial No. 10/185,940, filed June 27,

2002, both U.S. patent applications of which are incorporated herein by reference as if fully set out herein.

Third full paragraph beginning on line 29 of page 5 with:

The manifest 38 may include all components, including ~~dlls~~ dynamic-link libraries (dlls), component add-ins, Extensible Markup Language (XML), schema files and all associated XML files required by a software application for operating properly or required for improving, or adding, functionality to the software application 100. According to an embodiment of the present invention, the manifest 38 may also include information helpful to the end user, or to the end user's computer, for installing the downloaded components. For example, the manifest 38 may include information specifying the type for a given ~~dynamic-link library (dll)~~ so that the client-side computer 20 can properly register the dll including any need for specific registry keys for properly registering the dll.

Second full paragraph beginning on line 21 of page 10 with:

For example, if the user receives a document, such as a template, for the preparation of a resumé that "points" to a solution for providing helpful information, and actions, associated with completing the text and data fields of the template, the template or document received by the user may point to a remotely stored manifest 38 so that the user may download all files necessary for fully implementing the solution referred to by the document received by the user. For a detailed description of a method and system for creating, implementing and using "smart" documents such as the document 110, illustrated in Fig. 3, see ~~U.S. Patent Application entitled "*Providing Contextually Sensitive Actions and Help Content In Computer-Generated Documents,*" Serial No. _____, filed _____;~~ U.S. Patent Application entitled "*Providing contextually sensitive tools and help content in computer-generated documents,*" Serial No. 10/164,960, filed December 11, 2003, which is incorporated herein by this reference as if fully set out herein.

First full paragraph beginning on line 11 of page 12 with:

With reference to Fig. 2, an exemplary system for implementing the invention includes a conventional personal computer 20, including a processing unit 21, a system memory 22, and a system bus 23 that couples the system memory to the processing unit 21. The system memory 22 includes read-only memory (ROM) 24 and random access memory (RAM) 25. A basic input/output system 26 (BIOS), containing the basic routines that help to transfer information between elements within the personal computer 20, such as during start-up, is stored in ROM 24. The personal computer 20 further includes a hard disk drive 27, a magnetic disk drive 28, e.g., to read from or write to a removable disk 29, and an optical disk drive 30, e.g., for reading a CD-ROM disk 31 or to read from or write to other optical media. The hard disk drive 27, magnetic disk drive 28, and optical disk drive 30 are connected to the system bus 23 by a hard disk drive interface 32, a magnetic disk drive interface 33, and an optical disk drive interface 34, respectively. The drives and their associated computer-readable media provide non-volatile storage for the personal computer 20. Although the description of computer-readable media above refers to a hard disk, a removable magnetic disk and a CD-ROM disk, it should be appreciated by those skilled in the art that other types of media which are readable by a computer, such as magnetic cassettes, flash memory cards, digital video disks, Bernoulli cartridges, and the like, may also be used in the exemplary operating environment.

First full paragraph beginning on line 4 of page 14 with:

Fig. 3 is a computer screen display of a software application graphical user interface through which is displayed a document and associated contextually sensitive actions and help content according to an embodiment of the present invention. The document called "Microsoft Employee Performance Review" (reference numeral) ~~110~~ 112 illustrated in Fig. 3 represents a "smart" document that may point to functionality that may be downloaded from the manifest 38. The document ~~110~~ 112 illustrated in Fig. 3 is described for purposes of example only and is not limiting of the invention described herein. The word processing application 100 provides typical functionality associated

with a word processor accessible via drop down menus such as, File, Edit, View, Insert, Format, etc. The document ~~110~~ 112 is displayed in the work area of the application 100, and a document actions pane 135 is illustrated to the right of the document ~~110~~ 112.

Second full paragraph beginning on line 15 of page 14 with:

When the user places her computer cursor within a particular section of the document ~~110~~ 112, for example the "objectives" section 125 illustrated in Fig. 3, the user is provided with actions and help content in the document actions pane 135. For example, if the user places her computer cursor in the "objectives" section 125, the user may be provided with "Objective Writing Tips" 155 shown in the document actions pane 135. Selection of the "Objective Writing Tips" 155, as illustrated in Fig. 3, causes a display of "Objective Writing Tips" text 160 that provide the user with helpful information as to how to complete the "objectives" section in the performance review document ~~110~~ 112, illustrated in Fig. 3. If the user moves the cursor to a different section of the document, for example the personal information section 120, information provided in the document actions pane 135 will be provided to give the user assistance with the completion of the personal information section 120.

Third full paragraph beginning on line 27 of page 14 with:

In addition to helpful information to assist the user, a variety of document actions 145 are provided. For example, the "Submit Review" action may allow the user to submit the completed document ~~110~~ 112 to her supervisor or to her employee after completion of the document. The "Open Last Review" action may allow the user to open the last performance review so that she may determine whether she completed her objectives as set forth in the last review. If the document in use by the user is some other type of document, for example a resumé document, helpful information in the document actions pane might include information on preparing the "education" section, the "experience" section, and/or the "personal information" section.

First full paragraph beginning on line 6 of page 15 with:

According to the present invention, and as described in detail below, when a user focuses on a particular portion of the document ~~110~~ 112, such as the "objectives" section of the performance review ~~112~~ illustrated in Fig. 3, a solution property 115 points the document to the "objectives" section help solution illustrated in the document actions pane 135. The solution location 118 provides the document ~~110~~ 112 and the application 100 with the location of the components, dlls, or XML files necessary for implementing that solution. As should be understood, exemplary components may include components for drawing the document actions pane 135, components for displaying the information associated with the particular context, in this case the "objectives" section, and components for executing document actions such as "Submit Review" action 145. If the components needed by the application 100 and document ~~110~~ 112 to provide the "smart" functionality described above have not been downloaded to the user's computer 20, then these components may be downloaded to the computer 20 in accordance with the present invention as described below.

Third full paragraph beginning on line 26 of page 16 with:

If the document ~~110~~ 112 points to various components and files necessary for the operation of the application 100 and for editing the document ~~110~~ 112, or if the document points to a solution that may be downloaded to transform the document into a "smart" document, the method proceeds to step 420, and the application 100 checks the schema library 105 resident on the user's computer 20 for the presence of the necessary components or files. If the document includes solution properties pointing to more than one solution, the user of the document may be prompted (e.g. the dialog box 500, illustrated in Fig. 5) to select one of the solutions for downloading from the remote server.

Third full paragraph beginning on line 22 of page 21 with:

Once the document ~~110~~ 112 is opened after the download of component updates or additions, the user of the document now has all available functionality downloaded

onto the user's computer 20. If the document is a "smart" document as illustrated with reference to Fig. 3, the document will provide the user with all the normal functionality required for operating the document, ~~an~~ and additionally, the user will have useful help content and document actions based on the editing context within the document. For example, if the user has opened a ~~resume~~ resumé template, the user may receive help content and document actions based on the position of the user's computer cursor in a section of the document. In addition to pointing to the solution ID for obtaining the solution for use by the document, the document will point to the solution location from where the software components or software component upgrades may be downloaded to upgrade or add to the functionality in use by the document. Accordingly, with the present invention, the user may readily obtain downloaded updates and additions to the functionality available for use with the document being edited.

Second full paragraph beginning on line 14 of page 22 with:

As described above, with reference to Fig. 1, a manifest may be utilized for obtaining software components for enhancing a software application or for providing a solution to a document to make the document more intelligent or "smart." According to an alternate embodiment, a manifest may be utilized for dynamically configuring a document solution based on information obtained about the user of the document. Fig. 10 is a block diagram illustrating a system architecture for use in conjunction with an embodiment of the present invention showing interaction between a user's computer and a remotely maintained manifest and manifest collection. Referring to Fig. 10, according to this embodiment of the present invention, when a user obtains or creates a document that references or points to a document solution requiring that the solution or the components of the solution be obtained from the manifest 1038, a request is sent from the user's document via the user's computer 20 to the manifest 1038 via the server 49. According to this embodiment of the present invention, the manifest 1038 resides in an ~~active server page~~ Active Server Page 1000 having computer executable instructions sufficient for dynamically configuring a desired or required document solution based on information obtained about the user of the document solution.

Second full paragraph beginning on line 17 of page 25 with:

According to embodiments of the invention, the manifest collection 1040 contains a listing of namespace/solution pairs 1055 for directing a document to a document solution maintained at one or more manifests 1038. The manifest collection 1040 serves as a central repository for maintaining a listing of solutions for all the different namespaces referenced by documents maintained an[[d]] operated by the organization. On each user's computer 20, a registry key is designated for pointing the user's computer 20 to a uniform resource locator (URL) associated with the manifest collection. If the user opens a document referencing a document solution associated with one or more namespaces, the document may point to the URL of the manifest collection via the registry key set on the user's computer 20 and pass the name of a given document namespace to the manifest collection 1040.

First full paragraph beginning on line 15 of page 26 with:

Fig. 11 is a flowchart illustrating a method for obtaining a dynamically configured document solution for use in conjunction with a user document according to embodiments of the present invention. Referring to Fig. 11, an example operation of the manifest 1038 and manifest collection 1040 described above with reference to Fig. 10 will be described. The method 1100 begins at start block 1105 and proceeds to block 1110 where a user opens a document, for example, an employee review template, at the user's computer 20. At block 1115, ~~a determination is made as to~~ whether the document contains a solution ID and namespace associated with a document solution is checked. If so, a determination is made in following decision block 1120 as to whether a solution matching the solution ID and namespace is available in a local schema library 105 as described above with reference to Figs. 1 and 4.

Please delete the Sample Manifest Schema beginning on page 6, line 17 and ending on page 10, line 11.

Please insert the Sample Manifest Schema below after the last paragraph of the specification on page XX, before the claims:

Sample Manifest Schema

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="urn:schemas-microsoft-com:smartdocuments:manifest"
    targetNamespace="urn:schemas-microsoft-
com:smartdocuments:manifest" elementFormDefault="qualified">
  <xsd:annotation>
   <xsd:documentation xml:lang="en">
    The schema for office smart document, transform and schema manifests.
   </xsd:documentation>
  </xsd:annotation>
  <xsd:element name="manifest" type="manifestType" />
  <xsd:complexType name="manifestType" mixed="true">
    <xsd:sequence>
      <xsd:element name="version" type="xsd:string" />
      <xsd:element name="location" type="xsd:string" />
      <xsd:element name="updateFrequency"
type="zeroAndPositiveInteger" minOccurs="0" maxOccurs="1"/>
      <xsd:element name="uri" type="xsd:string" />
      <xsd:element name="manifestURL" type="xsd:string"
minOccurs="0" maxOccurs="1" />
      <xsd:element name="solution" type="solutionDefn"
minOccurs="1" maxOccurs="unbounded" />
    </xsd:sequence>
  </xsd:complexType>
  <xsd:complexType name="solutionDefn">
    <xsd:sequence>
      <xsd:element name="solutionID" type="xsd:string" />
      <xsd:element name="type" type="solutionType" />
      <xsd:element name="alias">
        <xsd:complexType>
          <xsd:simpleContent>
            <xsd:extension base="xsd:string">
              <xsd:attribute name="lcid" type="xsd:string"
use="required"/>
            </xsd:extension>
          </xsd:simpleContent>
        </xsd:complexType>
      </xsd:element>
      <xsd:element name="solutionSpecific" type="xsd:string"
minOccurs="0" maxOccurs="1" />
      <xsd:element name="file" type="fileType" minOccurs="1"
maxOccurs="unbounded"/>
```

```
            </xsd:sequence>
        </xsd:complexType>
        <xsd:simpleType name="solutionType">
            <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="schema" />
                    <xsd:enumeration value="transform" />
                    <xsd:enumeration value="smartDocument" />
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:complexType name="fileType">
              <xsd:all>
                    <xsd:element      name="runFromServer"      type="xsd:string"
minOccurs="0" maxOccurs="1"/>
                    <xsd:element name="type" type="typeDefn" />
                    <xsd:element      name="application"      type="xsd:string"
minOccurs="0" maxOccurs="1" />
                    <xsd:element name="version" type="xsd:string" />
                    <xsd:element  name="context"  type="xsd:string"  minOccurs="0"
maxOccurs="1" />
                    <xsd:element name="filePath" type="xsd:string" />
                    <xsd:element      name="installPath"      type="xsd:string"
minOccurs="0" maxOccurs="1" />
                    <xsd:element  name="register"  type="xsd:string"  minOccurs="0"
maxOccurs="1" />
                    <xsd:element  name="CLSID"  type="xsd:string"  minOccurs="0"
maxOccurs="1" />
                    <xsd:element  name="progID"  type="xsd:string"  minOccurs="0"
maxOccurs="1" />
                    <xsd:element name="regsvr32" type="xsd:string" minOccurs="0"
maxOccurs="1" />
                    <xsd:element  name="regasm"  type="xsd:string"  minOccurs="0"
maxOccurs="1" />
                    <xsd:element      name="registry"      type="registryType"
minOccurs="0" maxOccurs="1"/>
              </xsd:all>
        </xsd:complexType>
        <xsd:simpleType name="typeDefn">
            <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="template" />
                    <xsd:enumeration value="smartTagList" />
                    <xsd:enumeration value="solutionList" />
                    <xsd:enumeration value="schema" />
                    <xsd:enumeration value="transform" />
                    <xsd:enumeration value="actionHandler" />
                    <xsd:enumeration value="solutionActionHandler" />
```

```
                    <xsd:enumeration value="recognizer" />
                    <xsd:enumeration value="solutionRecognizer" />
                    <xsd:enumeration value="solutionList" />
                    <xsd:enumeration value="COMAddIn" />
                    <xsd:enumeration value="assembly" />
                    <xsd:enumeration value="XMLFile" />
                    <xsd:enumeration value="installPackage" />
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:complexType name="registryType">
            <xsd:all>
                    <xsd:element     name="registryKey"     type="registryKeyType"
minOccurs="1" />
            </xsd:all>
        </xsd:complexType>
        <xsd:complexType name="registryKeyType">
            <xsd:all>
                    <xsd:element name="keyName" type="xsd:string" />
                    <xsd:element name="keyValue" type="keyValueType" />
            </xsd:all>
        </xsd:complexType>
        <xsd:complexType name="keyValueType">
            <xsd:all>
                    <xsd:element name="valueName" type="xsd:string" />
                    <xsd:element name="valueType" type="registryValueType" />
                    <xsd:element name="value" type="xsd:string" />
            </xsd:all>
        </xsd:complexType>
        <xsd:simpleType name="registryValueType">
            <xsd:restriction base="xsd:string">
                    <xsd:enumeration value="REG_SZ" />
                    <xsd:enumeration value="REG_EXPAND_SZ" />
                    <xsd:enumeration value="REG_DWORD" />
            </xsd:restriction>
        </xsd:simpleType>
        <xsd:simpleType name="zeroAndPositiveInteger">
            <xsd:restriction base="xsd:integer">
                <xsd:minInclusive value="0"/>
            </xsd:restriction>
        </xsd:simpleType>
        </xsd:schema>
```